CLAIMS

1. A cryptographic method during which an integer division of the type $q = a \text{ div } b$ and $r = a \text{ mod } b$ is performed, with q a quotient, a a number of m bits, b a number of n bits with n less than or equal to m and b_{n-1} non-zero, b_{n-1} being the most significant bit of b, a method during which, at each iteration of a loop subscripted by i varying between 1 and m-n+1, a partial division of a word A of n bits of the number a by the number b is performed in order to obtain a bit of the quotient q,

the method being characterised in that the same operations are performed at each iteration, whatever the value of the quotient bit obtained.

2. A method according to Claim 1, during which, at each iteration, an addition of the number b to the word A and a subtraction of the number b from the word A are performed.

3. A method according to one of Claims 1 to 2, during which all the following steps are performed :

```
Input :  $a = (0, a_{m-1}, \ldots, a_0)$
         $b = (b_{n-1}, \ldots, b_0)$
Output:  $q = a \text{ div } b$ and $r = a \text{ mod } b$
$A = (0, a_{m-1}, \ldots, a_{m-n+1})$ ; $\sigma' \leftarrow 1$
For j = 1 to (m-n+1), do:
    $a \leftarrow SHL_{m+1}(a, 1)$ ; $\sigma \leftarrow carry$
    $A \leftarrow (\sigma')SUB_n(A, b) + (\neg\sigma')ADD_n(A, b)$
    $\sigma \leftarrow (\sigma' \text{ AND } \sigma') \ / \ (\sigma' \text{ AND } carry) / \ (\sigma' \text{ AND } carry)$
   $lsb(a)\ \ \sigma'$
   $\sigma' \leftarrow \sigma$
End For
if $(\neg\sigma = \text{TRUE})$ then $A \leftarrow ADD_n(A, b)$
```

4. A method according to Claim 1, during which, at each iteration, an operation of addition either of the number b or of a number $\bar{b}$ complementary to the number b with the word A is performed.

5. A method according to Claim 4, during which, at each iteration, an updating is also carried out of a first variable (σ′) indicating whether, during the following iteration, the number b or the number $\bar{b}$ must be added with the word A according to the quotient bit produced (lsb(a)).

6. A method according to Claim 4 or Claim 5, during which all the following steps are performed :

```
Input :  $a = (0, a_{m-1}, \ldots, a_0)$
         $b = (b_{n-1}, \ldots, b_0)$
Output:  q = a div b and r = a mod b
$A = (0, a_{m-1}, \ldots, a_{m-n+1})$ ; $\sigma'$ <- 1 ; $\bar{b}$ <- $\mathrm{CPL2}_N(b)$
For j = 1 to (m-n+1), do:
    a <- $\mathrm{SHL}_{m+1}(a, 1)$ ; $\sigma$ <- carry
    $d_{addr}$ <- $b_{addr} + \sigma' (\bar{b}_{addr} - b_{addr})$
    A <- $\mathrm{ADD}_n(A, d)$
    $\sigma$ <-($\sigma'$ AND $\sigma'$) / ($\sigma'$ AND carry)/ ($\sigma'$ AND carry)
    lsb(a) <- $\sigma'$
    $\sigma'$ <- $\sigma$
End For
if ($\neg\sigma$ = TRUE) then A <- $\mathrm{ADD}_n(A, b)$
```

7. A method according to Claim 1, during which, at each iteration, an operation of complement to 2^n of an updated data item (b or $\bar{b}$) or of a notional data item (c or $\bar{c}$) is performed, and then an operation of addition of the updated data item with the word A.

8. A method according to Claim 7, during which, at each iteration, an operation of updating a second variable

(δ) is also performed, indicating whether, during the following iteration, the operation of complement to 2^n must be performed on the updated data item or on the notional data item..

9. A method according to one of Claims 7 or 8, in which, at each iteration, there is also performed an updating of a third variable (β) indicating whether the updated data item is equal to the data item b or to its complement to 2^n.

10. A method according to one of Claims 7 to 9, during which all the following steps are also performed :

```
Input :  $a = (0, a_{m-1}, \ldots, a_0)$
         $b = (b_{n-1}, \ldots, b_0)$
Output:  $q = a \text{ div } b$ and $r = a \bmod b$
$\sigma' \leftarrow 1$ ; $\beta \leftarrow 1$, $\gamma \leftarrow 1$ ; $A = (0, a_{m-1}, \ldots., a_{m-n+1})$
for $j = 1$ to $(m-n+1)$, do:
    $a \leftarrow SHL_{m+1}(a, 1)$ ; $\sigma \leftarrow carry$
    $\delta \leftarrow \sigma' / \beta$
    $d_{addr} \leftarrow b_{addr} + \delta\ (c_{addr} - b_{addr})$
    $d \leftarrow CPL2_n(d)$
    $A \leftarrow ADD_n(A, b)$
    $\sigma \leftarrow (\sigma \text{ AND } \sigma') / (\sigma \text{ AND } carry) / (\sigma' \text{ AND } carry)$
    $\beta \leftarrow \neg\sigma'$ ; $\gamma \leftarrow \gamma / \delta$; $\sigma' \leftarrow \sigma$
    $lsb(a) = \sigma$
end for
if $(\neg\sigma = TRUE)$ then $A \leftarrow ADD_n(A, b)$
```

11. A method according to Claim 10, during which, at the end, the following operations are performed :

```
    if $(\neg\beta = TRUE)$ then $b \leftarrow CPL2_n(b)$
    if $(\neg\gamma = TRUE)$ then $c \leftarrow CPL2_n(c)$
```

12. An electronic component comprising calculation

means programmed to implement a method according to one of Claims 1 to 11, the calculation means comprising in particular a central unit associated with a memory comprising several registers for storing the data a and b.

13. A chip card comprising an integrated circuit according to Claim 12.